Temporal Knowledge Graph Reasoning based on Hierarchical Historical Contrastive Learning

Hongyan ZHAO, Weiqiang YUAN, Gaimei GAO
School of Computer Science and Technology, Taiyuan University of Science and Technology
hongyanzhao@tyust.edu.cn

Abstract

Temporal knowledge graph reasoning (TKGR) aims to use historical data to predict future facts. Most existing works have achieved some results by directly incorporating temporal information into static knowledge graph (SKG) embedding. However, they ignore the contextual information in the structure of the temporal knowledge graph (TKG). Moreover, the importance of different relations within each timestamp for predicting future facts has not been taken into account. How to comprehensively model the semantic relations of historical facts with different relations and the temporal information between facts is the difficulty of TKGR. To this end, this paper proposes a new temporal knowledge graph reasoning model based on hierarchical historical contrastive learning, called HHCLNET. Firstly, according to whether the relation with the query is the same, this model divides historical events into the same historical events (SHEs) and different historical events (DHEs), with corresponding entities called S-entities and D-entities. Then, S-entities and D-entities are analyzed and processed separately, and a graph attention mechanism is used to assign correlation scores for them. Next, using optimized contrastive learning methods, SHEs and DHEs are compared to obtain historical information that is truly relevant to the target query. Finally, a missing entity at future timestamp is predicted based on the two-layer historical learning results. Extensive experiments on five public available datasets demonstrate that the HHCLNET model has achieved significant improvements in performance. Especially, it achieves up to 8.7% improvement in MRR on GDELT for entity prediction comparing to the state-of-the-art baseline.

1 Introduction

Knowledge graphs have been widely used for natural language processing downstream tasks such as automated question and answer, dialogue or information retrieval due to their good knowledge storage capacity and reasoning ability. Traditional knowledge graphs are usually static knowledge graphs, which go about describing facts in the form of RDF triples (SHU et al., 2021), typically represented as (s, r, o), where s denotes the head entity, o denotes the tail entity, and r denotes the type of relation between them. In reality, the relational facts between entities are often time-sensitive, and the facts continue to change over time. Static-based knowledge graphs ignore the timeliness of entity-relation representations, fail to portray the evolutionary relations of dynamic facts, and the results predicted based on static knowledge graphs are usually not accurate enough and are limited in many tasks. For example, "Yao Ming played for the Houston Rockets of the NBA from 2002 to 2011", and the fact that Yao Ming played for the Rockets in 2017 no longer holds true. Moreover, time plays a very important role in some complex predictive and deductive reasoning tasks. To this end, TKGs have been proposed, which add temporal information to the factual representation, expanding the triple of SKG into the quadruple, which is represented as (s, r, o, t), where t denotes the temporal information, e.g., (Barack Obama, Campaign, President, 2012). TKG utilizes quadruples to dynamically represent facts, accurately capturing temporal semantic dependencies, expressing rich temporal semantic information (Liu et al., 2023), and achieving prediction of future temporal dynamic facts, which has important application value.

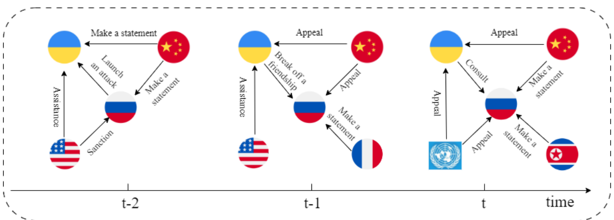

Figure 1: An example of a temporal knowledge graph

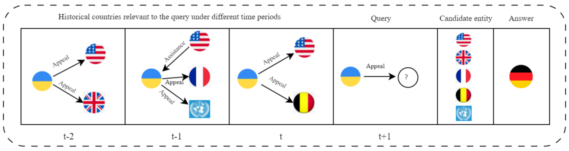

Figure 2: A reasoning example of temporal knowledge graph

TKGR (Mirtaheri et al., 2023) is the introduction of temporal information into knowledge representation and reasoning tasks, aiming to infer unknown information by existing historical information, which not only achieves the mining and completing of missing information of dynamic events, but also achieves the prediction of future events. It has already shown good application prospects in downstream tasks, such as stock prediction, public opinion monitoring, and transaction recommendation. However, the existing TKG, although large in volume, still suffers from the problem of missing and incomplete knowledge. How to effectively use the historical knowledge and temporal information in the knowledge base to infer unknown knowledge, and to supplement and extend the knowledge graph is still an urgent problem. Since temporal knowledge holds only for a fixed period of time and knowledge emerges new knowledge over time, TKGR is more challenging than traditional knowledge graph reasoning.

TKGR consists of two subtasks: entity prediction and relation prediction. This paper focuses on the former. For this task, the work RENET (Jin et al., 2020) and CyGNet (Zhu et al., 2021) try to solve the entity prediction task by modelling the historical events related to the subject entity in each query, but they neglect the treatment of the non-occurring historical entities and the impact of some non-significant entities on the prediction results within the same historical timestamp. In Figure 1, a segment of the development of the events of the conflict between Russia and Ukraine is shown for the target query (Ukraine, appeal, ?, 2022), the predicted event is a repeated event that has occurred in history. It can be predicted using historical event modelling, as Ukraine has previously asked for help from the United States in the historical event. In reality, Ukraine may turn to countries that have not been contacted before, i.e., the entity that is ultimately predicted has not appeared in historical events. As shown in Figure 2, this scenario is referred to as a new entity prediction. Presently, there are several temporal knowledge bases available, such as Wikidata (LEHMANN et al., 2015), Global Database of Events, Language, and Tone (GDELT) (SHEN et al., 2020), and Integrated Crisis Early Warning System (ICEWS) (WARD et al., 2012). When reasoning about the occurrence of future events based on these datasets, a significant portion of the events have few or no historical counterparts in history, which poses a significant challenge to the reasoning. For example, in the ICEWS database, new events that had never occurred before accounted for about 40%. Most existing methods focus on historical entities with a high frequency of past occurrences, and this extrapolation often leads to less accurate predictions.

To address the aforementioned challenges, this paper proposes the HHCLNET model, which divides the historical events into various historical subgraphs according to different time periods, aggregates the information representations of entities and relations in their domains through the multirelational neighbourhood aggregator, and then applies the Graph Attention Network to assign different weights to historical entities in each timestamp according to the different relations, and in Optimizing contrastive learning layer, the data are enriched by joining the relevant historical events, and the probability distributions of predicted entities are obtained at the end.

The contributions of the paper are summarized as follows:

- A new TKGR model based on hierarchical historical contrastive learning (HHCLNET) is proposed for predicting future missing entities. This model not only can predict high-frequency and repetitive events well, but also predict low-frequency and new events well.

- To the best of our knowledge, HHCLNET is the first to model the semantic relations of historical events with different relations and the temporal information between events, and GAT is used to fully consider the importance of different historical entities under multi-relations in the model.

- In order to better predict new facts that have not appeared in history, a historical contrastive learning method is proposed. This method optimizes and compares SHEs with DHEs, and learns the truly relevant historical events to the query.

- The performance of the model is validated on five publicly available datasets, and the experimental results show that the HHCLNET model obtains a large improvement in MRR, Hits@1, Hits@3, and Hits@10 compared to all baseline models.

2 Related Works

2.1 Static Knowledge Graph Reasoning

In the early days of knowledge graph research, static knowledge graph reasoning (SKGR) was focused, and it was an important task in knowledge graph completion. The TransE (BORDES et al., 2013) model based on SKGR regards relations as the translation of head entities to tail entities in vector space. It has a fast computation speed and is easy to implement, but it cannot solve the problems of one to many relations and many to one relations. To address these issues, variant Trans-series models have emerged, such as TransH (WANG et al., 2014), TransR (LIN et al., 2015), TransD (HE et al., 2015), etc., which solve the multisyntactic expression relation to a certain extent, but with high computational complexity. Later a new knowledge graph embedding model, RotatE (SUN et al., 2019), defines each relation as a rotation from the source entity to the target entity in the complex vector space, which greatly simplifies the computational complexity, but the model is sensitive to the data quality and the generalisation ability is unknown. Subsequently, matrix decomposition-based models ComplEx (Trouillon et al., 2016) and DistMult (Yang et al., 2015) were proposed. ComplEx (Trouillon et al., 2016) introduced the complex space into the knowledge graph embedding for the first time, while the DistMult (Yang et al., 2015) model defined the embedding of relations as diagonal matrices. Although the above models perform well in the embedded representation of the knowledge graph, the training time is long and the interpretability is not strong enough to explain the complex patterns between entities and relations in KGs.

2.2 Temporal Knowledge Graph Reasoning

TKGR adds time information to the SKG and achieves better inference performance. To address the embedding of temporal information, the TTransE (Leblay and Chekol, 2018) model adds time to the embedding of relations for inference, but does not explicitly capture entity-level temporal patterns, such as event periodicity. The RENET (Jin et al., 2020) model decomposes the joint probability distribution of relevant historical events into a series of conditional probability distributions and captures certain long-term dependencies, but ignores the problem of temporal variability, leading to inaccurate predictions of the final entity. Aiming at the lack of interpretability of the existing TKGR models, the xERTE (Han et al., 2021) model establishes the first interpretable time-associated attention prediction model, which is based on a new time-associated attention mechanism that preserves the causality of temporal multirelational data, but it is not sufficiently comprehensive to capture the local semantic information features of the entities in the TKG, and the identification of some important entities and the prediction of new entities are yet to be further investigated in depth. CyGNet (Zhu et al., 2021) combines Copy mode and Generation mode to predict new facts in the entire entity vocabulary using the historical vocabulary as a modulus. However, it not only ignores the value of non-negative frequency information, but also fails to take into account the problem of temporal variability in historical development. The RE-GCN (Li et al., 2021) model learns by modelling the evolution of historical sequences of a certain length, but ignores the problem of time variability in TKGR. The CEN (Li et al., 2022) model solves evolutionary patterns of different lengths by means of a course-learning strategy, but this approach requires constant cyclic training of the dataset, which greatly reduces the time efficiency of model training. The DA-Net (Liu et al., 2022) model first obtains repeated historical facts and then uses a combination of attentional mechanisms and frequency statistical information to solve the time-varying problem, but the statistical process of repeated historical facts and the attentional supervision process are both time-consuming .

3 Method

This section will focus on the HHCLNET model, and the model architecture is shown in Figure 3. The model consists of four modules: historical subgraph construction, analysis and processing of historical entities, optimizing contrastive learning and entity prediction module.

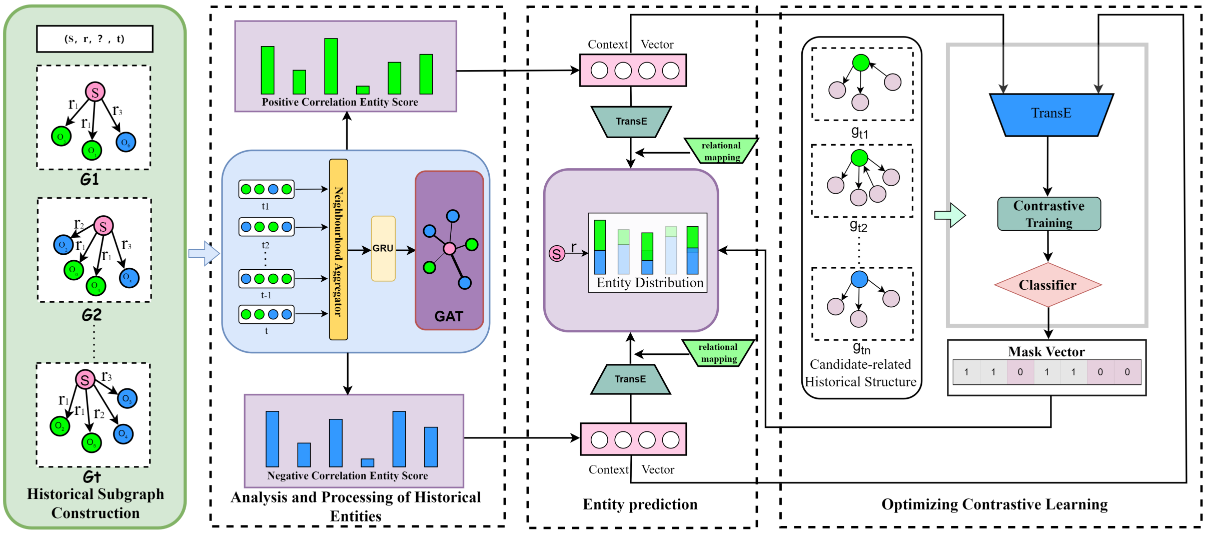

Figure 3: The HHCLNET model architecture. Firstly, the historical subgraph construction module generates historical subgraphs based on the query. Then, the analyzing and processing of historical entities module uses the graph attention mechanism to obtain the relevance scores of entities in each relation of the historical subgraph to the target query. Thirdly, the optimizing contrastive learning module compares SHEs to DHEs and adds historical events related to candidate entities to the comparison phase. Entity prediction module combines the previous two modules to generate the final result.

3.1 Historical Subgraph Construction

The module transforms the event and time context related to the query into a structured history subgraph. For the query q $(s, r, ?, t_n)$ or $(?, r, s, t_n)$, the historical subgraphs for each timestamp are obtained based on the known head entity s or tail entity o, and the event relations within each timestamp are multi-relational. In order to predict the missing entities in q, the historical entities within each timestamp are denoted as $\{\mathcal{X}_{t_i} \in \mathbb{R}^N | t_0 \leq t_i \leq t_n\}$ and the relations are denoted as r_i. Then, vectorize the historical entities and relations and send them to the next module for processing.

3.2 Analysis and Processing of Historical Entities

This module focuses on analysing and processing the historical entities and corresponding relations within each timestamp. Firstly, a multi-relational neighbourhood aggregator is used to aggregate the entity information within the same timestamp. Secondly, the history information of each timestamp is fed into the GRU encoder to learn the dynamic features of the event evolution, and then each relation is assigned an attention weight through the graph attention network (GAT). Finally, the history entities are classified into positively and negatively correlated entities, and the corresponding scores are computed.

Multi-Relational Neighbourhood Aggregator. Since the entities within each timestamp are multi-relational, a neighbourhood aggregator is first used to aggregate the multi-relation neighbourhood entity features at the same time, further obtaining the graph representation of the target entity e_i, as shown in equation 1:

$$h_i = \sigma(\sum_{r \in R} \sum_{o \in \mathrm{N}_t} \frac{1}{C_s} W_r{}^l h_o{}^l + W_o{}^l h_s{}^l) \quad (1)$$

where N_t denotes the set of neighboring nodes of the target entity s in relation r at timestamp t, and C_s denotes the number of edges in the graph of the target entity s at the timestamp, which is used here as a normalization factor. h_o^l and h_s^ldenote the trainable embedding of entities e_o and e_s respectively. l denotes the number of aggregation layers. W_r^l and W_o^l are the learnable weight matrices, and $\sigma(\cdot)$ is the activation function RELU.

GRU components. According to the characteristics of temporal variability, the entities will continuously update and change, and the corresponding frequency will also change. Therefore, GRU (CHUNG et al., 2014) component is used to record the changes of neighboring entities to further enhance learning ability, as shown in equation 2:

$$e_i = GRU(\mathcal{X}_{ro,t_1}, \mathcal{X}_{ro,t_2}, \ldots, \mathcal{X}_{ro,t_n}) \quad (2)$$

Graph Attention Network. Here, the historical information obtained by GRU is input into the GAT to assign different weights to neighboring entities of different relations. Some important nodes will receive higher weights, thereby alleviating the impact of non important facts on neighborhood. The attention weight is calculated as follows:

$$g_{ijk} = \beta^T \sigma(h_{e_i} \cdot \varphi(e_j, r_k)) \tag{3}$$

where $\beta^T \in \mathbb{R}^d$ is the parameter vector, $\cdot$ denotes the element multiplication symbol, and r_k is the relation between the target head entity and the tail entity.

Score of positive and negative related entities. Historical entities are divided into positive and negative related entities based on whether the relations is the same as the query. In order to calculate the correlation score of historical entities, the frequency of each historical entity is counted. As shown in equation 4:

$$C_t^{(s,r)} = c_{t-n}^{(s,r)} + c_{t-n+1}^{(s,r)} + ... + c_{t-1}^{(s,r)} \tag{4}$$

where $C_t^{(s,r)}$ represents the number of times the entity has appeared in the history and $c_{t-n}^{(s,r)}$ is the number of times the entity has appeared in different time. Since historical facts are multi-relational within each timestamp, we assign positive correlation scores to S-entities, and negative correlation scores to D-entities. The positive correlation score is calculated by formula 5:

$$H_{positive}^{(s,r)} = tanh(W_1(s \oplus r) + b_1)E^T + C_t^{(s,r)} + g_{ijk} \tag{5}$$

where $tanh$ is the activation function, $\oplus$ represents the connection symbols, $W_1 \in \mathbb{R}^{d \times 2d}$ is the trainable weights, and $b_1 \in \mathbb{R}^d$ is the trainable bias. Adding bias here can play a stabilizing role in handling missing entities of different events, which is very necessary. Then we multiply the output of the linear layer by the E vector and add the frequency $C_t^{(s,r)}$ and the entity attention scores g_{ijk}, thus assigning higher scores to the relevant entities and obtaining more accurate attention scores. Negatively correlated entity scores are calculated by formula 6:

$$H_{negative}^{(s,r)} = tanh(W_2(s \oplus r) + b_2)E^T + C_t^{(s,r)} + g_{ijk} \tag{6}$$

Finally, the positive correlation entities and relations, and negative correlation entities and relations are vectorized for representation, and passed to softmax to get the probability of candidate entities, which is calculated as shown in equation 7, equation 8:

$$P_1 = softmax(H_{positive}^{(s,r)}) \tag{7}$$

$$P_2 = softmax(H_{negative}^{(s,r)}) \tag{8}$$

P_1 and P_2 are the probabilities of positively and negatively correlated entities, respectively. Entities with higher probability values are more correlated with predicted entities.

3.3 Optimizing Contrastive Learning

Over time, new events that have not appeared in history or have a lower frequency in history may arise. It requires a fuller understanding of the historical contextual information, not only from the set of positively correlated entities but also from the set of negatively correlated entities to discover entities related to the query. Moreover, existing models generally suffer from the data sparsity problem, leading to poor learning performance. Therefore, this module adopts an optimized contrastive learning method to compare and contrast the positively and negatively correlated historical information, and to identify the historical entities that are truly correlated and uncorrelated with the query.

Firstly, through the TransE embedding method, the positively related entities and relations, negatively related entities and relations, and the frequency of their respective occurrences in the history are represented, so as to obtain richer historical information. The TransE knowledge representation is used here to better model similarities knowledge and improve the reasoning accuracy. Let I_q be the embedded representation of the query information:

$$I_q = TransE(s \oplus r \oplus tanh(W_c C_t^{(s,r)})) \tag{9}$$

The sequence of historical subgraphs is defined here as $\{g_{t_1}^{e_j}, g_{t_2}^{e_j}, ..., g_{t_n}^{e_j}\}$, n is the maximum length of the sequence, and each subgraph is multi-relational. Firstly, the query is projected onto the plane by TransE, then the positively related entities are used as a positive sample and the negatively related entities are used as a negative sample in comparison training. These related historical events are added to enrich the positive and negative sample data.

Next, the definitions are given: minbatch is denoted as M, and the set with the same relation to query q is defined as $Q_{(q)}$. The identification of whether to focus on historical or new entities is done by minimizing the contrast loss function. The specific loss function is shown in equation 10:

$$\mathcal{L}^{con} = \sum_{q \in M} \frac{-1}{|Q(q)|} \sum_{k \in Q(q)} \log \frac{\exp(I_q \cdot I_k/\tau)}{\sum_{i=0} \exp(I_q \cdot I_i/\tau)} \tag{10}$$

where $\tau \in \mathbb{R}^+$ is the temperature parameter, being set to 0.1 here. After the data is enhanced by the comparison samples, it minimizes the $\mathcal{L}^{con}$ loss function, effectively modeling the characteristics of related samples, which helps to better model the semantic relatedness between related entities in the representation learning and improves the model's representational and inference capabilities.

Next, a binary classifier is used to output scalars between 0 and 1. Here, we set I_q greater than or equal to 0.5 to indicate that the prediction tends towards positively correlated entities, and I_q less than 0.5 to indicate that attention should be paid to negatively correlated entities. Finally, a masking strategy is used to process the predicted entity. Here we denote it by $Z_t^{s,r}(o) \in \mathbb{R}^{|\varepsilon|}$ vector. If the positively correlated entities are focused, $Z_t^{s,r}(o)$ is set to 1 for the positions of all positively correlated entities, and $Z_t^{s,r}(o)$ is set to 0 for the positions of all negatively correlated entities. In other words, if the missing entity is predicted to be in SHEs, then S-entities set will receive more attention. The reverse is true, too.

3.4 Entity prediction

In order to enhance the learning ability of the model, this module combines the probability obtained from the analysis and processing module of relevant historical entities with the optimizing contrastive learning module to obtain the probability distribution of the correlated entity. The probabilities of positively correlated entities P_1 and negatively correlated entities P_2 will be summed and averaged to obtain the probability$P_t^{s,r}$. Finally, $P_t^{s,r}$ will be multiplied with the vector $Z_t^{s,r}(o)$ to obtain the predicted probability of candidate entities. The entity with the highest probability will be selected as the final predicted entity.

$$P(o|s, r, C_t^{(s,r)}) = P_t^{s,r}(o) \cdot Z_t^{s,r}(o) \tag{11}$$

3.5 Training Strategy

The training process of the model mainly includes four steps. Firstly, HHCLNET searches for all historical events related to entity s for a given query (s, r,?, t). Sencondly, the model performs relation processing on relevant entities within different timestamps, generates a set of positively correlated and negatively correlated candidate entities, and uses a GAT to assign correlation scores to different entities. Thirdly, by increasing the data of positive and negative samples during the contrastive learning layer, a reasonable pair of positive and negative samples is selected for training. Finally, Combining the above two steps to obtain the contextual representation of the predicted entity, and probability distribution of candidate entity is obtained through a binary classifier and masking strategy.

Finally the model parameters are trained by the cross-entropy loss function.

$$\mathcal{L} = -\sum_{(s,r,o) \in G} \log p(o|s, r) + \lambda_1 \mathcal{L}^{con} \tag{12}$$

where G represents the entire history event, $p(o|s, r)$ denotes the probability of candidate entity o based on the given entity s and the relation r, and λ_1 is the weight coefficient.

4 Experiments

4.1 Datasets and Metrics

Dataset	Entities	Relation	Training	Validation	Test	Time gap
ICEWS18	23 033	256	373 018	45 995	49 545	24 hours
ICEWS14	7 128	260	63 685	—	13 222	24 hours
GDELT	7 691	240	1 734 399	238 765	305 241	24 hours
WIKI	12 554	24	539 286	67 538	63 110	1 year
YAGO	10 623	10	161 540	19 523	20 026	1 year

Table 1: Statistics information of datasets

To evaluate the method proposed in this paper, five commonly used benchmark datasets are used: ICEWS (including ICEWS14 and ICEWS18), YAGO, WIKI and GDELT. The ICEWS14 and ICEWS18 datasets divide each timestamp in 24-hour intervals. The ICEWS14 dataset collects events that occurred from 1 January 2014 to 31 December 2014, and the ICEWS18 dataset collects events from 1 January 2018 to 31 December 2018. The YAGO dataset collected from 2013 to 2017. The WIKI dataset is extracted from the Wikipedia database, which collects data from 2008 to 2017. During the experimental evaluation, the dataset is divided into training, validation and test sets by timestamps, which are 80%, 10% and 10%, respectively. We set the training batch size to 1024, the

Method	ICEWS18				ICEWS14				GDELT			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TransE	17.56	2.48	26.95	43.87	18.65	1.12	31.34	47.07	16.05	0.00	26.10	42.29
DisMult	22.16	12.13	26.00	42.18	19.06	10.09	22.00	36.41	18.71	11.59	20.05	32.55
CompIEX	30.09	21.88	34.15	45.96	24.47	16.13	27.49	41.09	22.77	15.77	24.05	36.33
R-GCN	23.19	16.36	25.34	36.48	26.31	18.23	30.43	45.34	23.31	17.24	24.96	34.36
ConvE	36.67	25.81	39.80	50.69	40.73	33.20	43.92	54.35	35.99	27.05	39.32	49.44
HyTE	7.31	3.10	7.50	14.95	11.48	5.64	13.04	22.51	6.37	0.00	6.72	18.63
TTransE	8.36	1.94	8.71	21.93	6.35	1.23	5.80	16.65	5.52	0.47	5.01	15.27
TeMp	40.48	33.97	42.63	52.38	43.13	35.67	45.79	56.12	37.56	29.82	40.15	48.60
RE-NET	42.93	36.19	45.47	55.80	45.71	38.42	49.06	59.12	40.12	32.43	43.40	53.80
RE-GCN	32.78	24.99	35.54	48.01	32.37	24.43	35.05	48.12	29.46	21.74	32.01	43.62
CyGNet	46.69	40.58	49.82	57.14	48.63	41.77	52.50	60.29	50.29	44.53	54.69	60.99
EvoKG	29.67	12.92	33.08	58.32	18.30	6.30	19.43	39.37	11.29	2.93	10.84	25.44
HGAT	28.55	19.68	32.74	46.60	46.68	29.72	42.46	56.45	39.12	26.35	45.31	56.62
GLANET	27.54	17.90	31.20	46.57	38.06	27.97	42.92	57.65	38.93	26.48	43.62	61.36
RPC	34.91	24.34	38.74	55.89	44.55	34.87	49.80	65.08	22.41	14.42	24.36	38.33
HIP	48.37	43.51	51.32	58.49	50.57	45.73	**54.28**	61.65	52.76	46.35	55.31	**61.87**
HHCLNET	**53.08**	**49.15**	**53.97**	**60.53**	**55.08**	**51.15**	54.02	**62.53**	**59.35**	**55.05**	**60.03**	60.35

Table 2: Experimental results of models in ICEWS18, ICEWS14 and GDELT

embedding dimension of entities and relations to 200, the learning rate to 0.001, the dropout to 0.5 to prevent overfitting, and the Adam optimizer is used for parameter optimization. We set the training epoch size to 30, the test epoch size to 20, and the validation epoch size to 10. The statistical information for the datasets is shown in Table 1.

The evaluation metrics generally used for TKGR are Mean Reciprocal Ranks (MRR) and the hit rate Hits@K for results in the top K. In this experiment, $Hits@1$, $Hits@3$, and $Hits@10$ are chosen as the evaluation metrics.

4.2 Baselines and Results

In order to validate the effectiveness of this model, comparisons are made among 16 baseline models, which can be classified into two types: SKGR methods, including TransE (BORDES et al., 2013), DistMult (Yang et al., 2015), CompIEX (Trouillon et al., 2016), R-GCN (Schlichtkrull et al., 2018), and ConvE (Dettmers et al., 2018); TKGR methods, including HyTE (DASGUPTA et al., 2018), TTransE (Leblay and Chekol, 2018), TeMp (Wu et al., 2020), RE-NET (Jin et al., 2020), RE-GCN (Li et al., 2021), CyGNet (Zhu et al., 2021), EvoKG (Park et al., 2022), HGAT (Shao et al., 2023), GLANET (Wang et al., 2023), RPC (Liang et al., 2023) and HIP (He et al., 2024). The entity prediction results of the above different models on the five datasets are given in Table 2 and Table 3, respectively. The experimental results show that the HHCLNET model has achieved the best performance on most metrics in all datasets. Especially on the WIKI and YAGO datasets, the performance improvement is particularly significant, respectively MRR and Hit@3 improved by 3.15% and 2.81% compared to the best baseline. Compared to this, the improvement on the ICEWS18, ICEWS14, and GDELT datasets is slightly smaller, because both the ICEWS and GDELT datasets are event-based datasets containing more complex relational networks and large amounts of data, which makes the processing slower and the probability of new events is high. However, the WIKI and YAGO datasets have a temporal granularity of years, fewer types of relations and most of them remain constant, with a high proportion of repetitive events. Therefore, when predicting entities on the WIKI and YAGO datasets, the factual relations that can be relied on are relatively stable, making the model's improvement effect significant. From the above experimental results, it can be seen that the HHCLNET model has significant effects on all datasets, and it also indicates that the model has indeed learned historical information related to the prediction in the historical entity analysis and processing module and optimizing contrastive learning module. The experimental results above indicate that compared to other baseline models, the model exhibits strong robustness and generalization when dealing with complex and noisy datasets. Detailed analysis can be found in the case studies provided in the appendix.

4.3 Ablation Study

In order to validate the importance of each module of the HHCLNET model, ablation experiments were carried out by keeping the experimental setup

Method	WIKI				YAGO			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TransE	46.68	36.19	49.71	51.71	48.97	46.23	62.45	66.05
DisMult	46.12	37.24	49.81	51.38	59.47	52.97	60.91	65.26
CompIEX	47.84	38.15	50.08	51.39	61.29	54.88	50.08	51.39
R-GCN	37.57	28.15	39.66	41.90	41.30	32.56	44.44	52.68
ConvE	47.57	38.76	50.10	50.53	62.32	56.19	63.97	65.60
HyTE	43.02	44.16	45.12	49.49	23.16	39.73	45.74	51.94
TTransE	31.74	35.36	36.25	43.45	32.57	26.10	43.39	53.37
TeMp	49.61	46.96	50.24	52.13	62.25	55.39	64.63	66.02
RE-NET	51.97	48.01	52.07	53.91	65.16	63.29	65.63	68.08
RE-GCN	44.86	39.82	46.75	47.56	65.69	59.98	68.70	69.22
CyGNet	45.50	50.48	50.79	52.80	63.47	64.26	65.71	68.95
EvoKG	50.66	12.21	63.84	68.03	55.11	54.37	81.38	79.65
HGAT	56.12	52.90	58.16	61.82	63.62	59.80	66.02	71.58
GLANET	53.18	58.23	61.16	71.52	65.05	76.32	77.86	79.24
RPC	65.31	67.82	69.73	70.23	84.71	83.82	82.73	85.23
HIP	64.71	63.82	68.73	58.23	77.55	76.32	78.49	80.23
HHCLNET	**68.46**	**70.35**	**71.44**	**71.66**	**85.53**	**85.28**	**85.54**	**85.84**

Table 3: Experimental results of models in WIKI and YAGO

constant and creating variants by adjusting the different components of the model, and ICEWS18 and YAGO datasets were chosen to carry out the experiments. The results of the experiments are shown in Table 4.

Ablation	ICEWS18				YAGO			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
HHCLNET-GAT	48.71	47.04	49.93	53.91	84.46	84.33	84.52	84.65
HHCLNET-OCon	52.09	48.21	51.92	58.78	84.79	85.35	85.02	84.49
HHCLNET	53.08	49.15	53.97	60.53	85.53	85.28	85.54	85.84

Table 4: Results of ablation study in ICEWS18 and YAGO

Here ICEWS18 and YAGO are chosen to investigate the effectiveness of graph attention network (GAT) and optimizing contrast learning (OCon). Table 4 shows the result of ablation. HHCLNET-GAT only cosiders the GAT module without OCon, while HHCLNET-OCon only keeps OCon module. From the experimental results, it can be seen that all two modules play a significant role in the model. The graph attention network makes the model fully consider the degree of importance of different neighborhood entities under multi-relations, which helps the model to obtain a more accurate probability distribution of entities. Optimizing contrastive learning can improve the model performance, reflecting the importance of selecting positive and negative samples in contrastive learning. And it can further strengthen the learning ability of the model and enhance the reasoning ability of the model.

4.4 Hyper-parameter Analysis

In order to assess the sensitivity of the HHCLNET model to the parameters, an experimental comparison of two parameters (batch size and Dropout) was performed on the dataset YAGO, where the batch size was set to {64, 128, 256, 512, 1024}, and the Dropout was set to {0.1, 0.3, 0.5, 0.7, 0.9}. As can be seen from Figure 4 and Figure 5, when the batch size and Dropout are (1024, 0.5) on the YAGO data set, the model achieved the best performance. This demonstrates that the HHCLNET model is sensitive to pairwise batch size and Dropout.

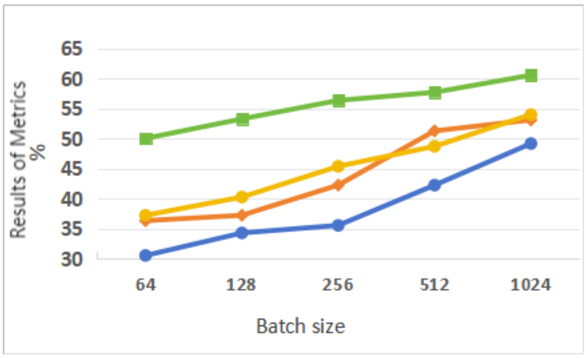

Figure 4: Batch training size on YAGO

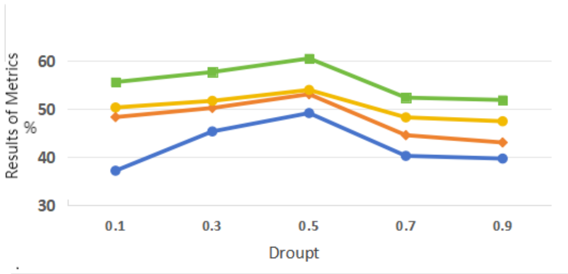

Figure 5: Droupout on YAGO

5 Conclusion and Future Work

In this paper, we proposes a new temporal knowledge graph reasoning model based on hierarchical historical contrastive learning (HHCLNET). The model analyses and processes the acquired historical entities, and then uses optimizing comparative learning to further identify truly relevant entities with the query, allowing the model to focus more on the useful entities. Moreover, the model has shown good performance in predicting new events, high-frequency events, and low-frequency events. Therefore, the model has good generalization ability. In subsequent research, we will work on fusing multi-source information to enhance the entity feature representation and thus continuously strengthen the learning capability of the model.

References

Shu, S., Li, S., Hao, X., & Zhang, L. (2021).

Knowledge graph embedding technology: A review. J. Front. Comput. Sci. Technol, 15, 2048.

Liu KZ, Zhao F, Jin H. FS-Net: Frequency Statistical Network for Temporal Knowledge Graph Reasoning. Ruan Jian Xue Bao/Journal of Software, 2023, 34(10): 4518-4532.

Mirtaheri, M., Rostami, M., & Galstyan, A. (2023). History repeats: Overcoming catastrophic forgetting for event-centric temporal knowledge graph completion. arXiv preprint arXiv:2305.18675.

Jin, W., Qu, M., Jin, X., & Ren, X. (2019). Recurrent event network: Autoregressive structure inference over temporal knowledge graphs. arXiv preprint arXiv:1904.05530.

Zhu, C., Chen, M., Fan, C., Cheng, G., & Zhang, Y. (2021, May). Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In Proceedings of the AAAI conference on artificial intelligence (Vol. 35, No. 5, pp. 4732-4740).

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., ... & Bizer, C. (2015). Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. Semantic web, 6(2), 167-195.

Shi, S., Changqing, S., Changxiu CHENG, J. G., & Sijing, Y. E. (2020). GDELT: Big event data for sensing global social dynamics. World Regional Studies, 29(1), 71.

Ward, M. D., Metternich, N. W., Carrington, C., Dorff, C., Gallop, M., Hollenbach, F. M., ... & Weschle, S. (2012, July). Geographical models of crises: Evidence from ICEWS. In 2nd International Conference on Cross-Cultural Decision Making: Focus (pp. 21-25).

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. Advances in neural information processing systems, 26.

Wang, Z., Zhang, J., Feng, J., & Chen, Z. (2014, June). Knowledge graph embedding by translating on hyperplanes. In Proceedings of the AAAI conference on artificial intelligence (Vol. 28, No. 1).

Yang, B., Yih, W. T., He, X., Gao, J., & Deng, L. (2014). Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575.

Sun, Z., Deng, Z. H., Nie, J. Y., & Tang, J. (2019). Rotate: Knowledge graph embedding by relational rotation in complex space. arXiv preprint arXiv:1902.10197.

Lin, Y., Liu, Z., Sun, M., Liu, Y., & Zhu, X. (2015, February). Learning entity and relation embeddings for knowledge graph completion. In Proceedings of the AAAI conference on artificial intelligence (Vol. 29, No. 1).

Ji, G., He, S., Xu, L., Liu, K., & Zhao, J. (2015, July). Knowledge graph embedding via dynamic mapping matrix. In Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers) (pp. 687-696).

Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., & Bouchard, G. (2016, June). Complex embeddings for simple link prediction. In International conference on machine learning (pp. 2071-2080). PMLR.

Leblay, J., & Chekol, M. W. (2018, April). Deriving validity time in knowledge graph. In Companion proceedings of the the web conference 2018 (pp. 1771-1776).

Han, Z., Chen, P., Ma, Y., & Tresp, V. (2020, October). Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In International Conference on Learning Representations.

Li, Z., Jin, X., Li, W., Guan, S., Guo, J., Shen, H., ... & Cheng, X. (2021, July). Temporal knowledge graph reasoning based on evolutional representation learning. In Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval (pp. 408-417).

Li, Z., Guan, S., Jin, X., Peng, W., Lyu, Y., Zhu, Y., ... & Cheng, X. (2022). Complex evolutional pattern learning for temporal knowledge graph reasoning. arXiv preprint arXiv:2203.07782.

Liu, K., Zhao, F., Chen, H., Li, Y., Xu, G., & Jin, H. (2022, October). Da-net: Distributed attention network for temporal knowledge graph reasoning. In Proceedings of the 31st ACM International Conference on Information & Knowledge Management (pp. 1289-1298).

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

Yang, J., Zhou, W., Wei, L., Lin, J., Han, J., & Hu, S. (2020). RE-GCN: relation enhanced graph convolutional network for entity alignment in heterogeneous knowledge graphs. In Database Systems for Advanced Applications: 25th International Conference, DASFAA 2020, Jeju, South Korea, September 24–27, 2020, Proceedings, Part II 25 (pp. 432-447). Springer International Publishing.

Han, Z., Chen, P., Ma, Y., & Tresp, V. (2020, October). Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In International Conference on Learning Representations.

Ward, M. D., Beger, A., Cutler, J., Dickenson, M., Dorff, C., & Radford, B. (2013). Comparing GDELT and ICEWS event data. Analysis, 21(1), 267-297.

Dettmers, T., Minervini, P., Stenetorp, P., & Riedel, S. (2018, April). Convolutional 2d knowledge graph embeddings. In Proceedings of the AAAI conference on artificial intelligence (Vol. 32, No. 1).

Dasgupta, S. S., Ray, S. N., & Talukdar, P. (2018). Hyte: Hyperplane-based temporally aware knowledge graph embedding. In Proceedings of the 2018 conference on empirical methods in natural language processing (pp. 2001-2011).

Wu, J., Cao, M., Cheung, J. C. K., & Hamilton, W. L. (2020). Temp: Temporal message passing for temporal knowledge graph completion. arXiv preprint arXiv:2010.03526.

He, Y., Zhang, P., Liu, L., Liang, Q., Zhang, W., & Zhang, C. (2024). HIP network: historical information passing network for extrapolation reasoning on temporal knowledge graph. arXiv preprint arXiv:2402.12074.

Park, N., Liu, F., Mehta, P., Cristofor, D., Faloutsos, C., & Dong, Y. (2022, February). Evokg: Jointly modeling event time and network structure for reasoning over temporal knowledge graphs. In Proceedings of the fifteenth ACM international conference on web search and data mining (pp. 794-803).

Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., & Welling, M. (2018). Modeling relational data with graph convolutional networks. In The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15 (pp. 593-607). Springer International Publishing.

Liang, K., Meng, L., Liu, M., Liu, Y., Tu, W., Wang, S., ... & Liu, X. (2023, July). Learn from relational correlations and periodic events for temporal knowledge graph reasoning. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 1559-1568).

Shao, P., He, J., Li, G., Zhang, D., & Tao, J. (2023). Hierarchical graph attention network for temporal knowledge graph reasoning. Neurocomputing, 550, 126390.

Wang, J., Lin, X., Huang, H., Ke, X., Wu, R., You, C., & Guo, K. (2023). GLANet: temporal knowledge graph completion based on global and local information-aware network. Applied Intelligence, 53(16), 19285-19301.

A Case Study

To further demonstrate the effectiveness of the proposed model, a relevant case study was conducted. As shown in Figure 6, we selected three representative queries from the ICEWS dataset to analyze the prediction results of HHCLNET.

· When the query is (Russia, visit, ?, t), it has not appeared in related historical events. The model analyzes negatively correlated entities through an optimization and comparison stage, predicting a high likelihood of such entities in China, with results consistently matching the correct answers. This indicates the model's ability to predict the correct entities that do not appear in the same historical relationship events.

· When given a query (US, invitation, ?, t), it can be observed from the graph that Canada has the highest probability of occurrence and belongs to a historical entity. The historical entity analysis processing module of the model assigns high correlation scores through graph attention, thus selecting the Canadian entity with the highest probability as the final prediction result.

· When given a query (United States, Cooperation, ?, t), as the relationship "US and Japan cooperation" appeared once in history, belonging to the positively correlated historical entities, the model combines the historical entity analysis module and the optimization and comparison learning module to get the final entity prediction result of Japan. The final prediction matched the correct answer. So the model's predictions are correct.

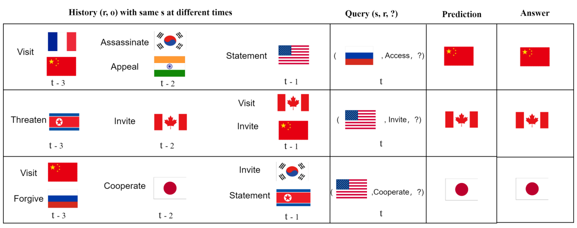

Figure 6: Case study of HHCLNET's predictions.

From the above cases, it can be seen that the hierarchical historical contrastive learning method proposed in this paper enables the model to automatically learn and query truly relevant historical events and candidate entities when facing tasks such as predicting new events, low-frequency event forecasting, and high-frequency event forecasting. The cases demonstrate that identifying useful entities helps improve reasoning outcomes, further proving the strong generalization ability of the proposed model.